#### Prof. Mrudula Mukadam



### **BIG DATA TECHNOLOGY**

Assignment 4



March 26, 2020 Ahmed Hamdy 611114

# Q1. Write an in-mapper combiner algorithm for the "average problem". (Pseudo code only; show reducer method too.)

Pseudo-code for the average problem with the in mapper combining:

```
class Mapper
       method setup ():
              memory ← new HashMap
       method map (String t, int r):
              if memory doesn't have key t:
                      add <t, pair (r, 1)> to memory
              else:
                     pair (s, c) \leftarrow get value of t from memory
                     add \langle t, pair(s + r, c + 1) \rangle to memory
       method cleanup ():
              for all \langle t, pair(s, c) \rangle in memory do:
                     Emit (t, pair (s, c))
class Reducer
       method reduce (String t, pairs [(s1, c1), (s2, c2), ...]):
              sum \leftarrow 0
              count \leftarrow 0
              for all pair (s, c) in pairs [(s1, c1), (s2, c2), ...] do:
                     sum \leftarrow sum + s
                     count \leftarrow count + c
              average ← sum / count
              Emit (t, average)
```

#### Q2. Table question with part a, b, c

Assume that there are three reducers. Note that Reducer 1 runs on Machine1. Reducer 2 runs on Machine2. Reducer 3 runs on Machine3. Further, let the partitioner assign all words starting from letter 'a-j' to Reducer 1, all words starting from letter 'k-q' to reducer 2 and everything else to Reducer 3.

Also assume that there are six input splits as follows:

Input split1: [cherry mango olive cherry]

[plum cherry banana cherry]

Input split2: [cherry banana radish radish]

[carrot banana mango cherry]

Input split3: [banana kiwi plum banana]

[mango cherry kiwi banana]

Input split4: [apple mango carrot plum]

[radish kiwi banana olive]

Input split5: [olive banana radish kiwi]

[cherry kiwi olive cherry]

Input split6: [banana radish plum banana]

[olive cherry banana radish]

Input splits 1, 2 are on Machine 1, input splits 3, 4 are on Machine 2 and input splits 5, 6 are on Machine 3.

- a) Illustrate the word count algorithm with no combiner, no in-mapper combining. show mapper o/p, reducer i/p and reducer o/p
- b) Illustrate the word count algorithm with combiner, no in-mapper combining. show mapper o/p, combiner o/p, reducer i/p and reducer o/p
- c) Illustrate the word count algorithm within mapper combiner. show mapper o/p, reducer i/p and reducer o/p

Remember to show the sorted mapper output that gets stored locally.

Note: Illustrate means show mapper o/p, combiner o/p (if using combiners), reducer i/p and reducer o/p.



#### a) No combiner, no in mapper combining

Machine 1		Machine 2		Machine 3	
Mapper 1 – Input Split 1- output		Mapper 3 – Input Split 3-		Mapper 5 – Input Split 5-	
Wiapper 1 Imput Split 1- Output		output		output	
<cherry, 1=""></cherry,>	<plu><plum, 1=""></plum,></plu>	<banana, 1=""></banana,>	<mango, 1=""></mango,>	<olive, 1=""></olive,>	<cherry, 1=""></cherry,>
<mango, 1=""></mango,>	<cherry, 1=""></cherry,>	<kiwi, 1=""></kiwi,>	<cherry, 1=""></cherry,>	<banana, 1=""></banana,>	<kiwi, 1=""></kiwi,>
<olive, 1=""></olive,>	<banana, 1=""></banana,>	<plu><plum, 1=""></plum,></plu>	<kiwi, 1=""></kiwi,>	<radish, 1=""></radish,>	<olive, 1=""></olive,>
<cherry, 1=""></cherry,>	<cherry, 1=""></cherry,>	<banana, 1=""></banana,>	<banana, 1=""></banana,>	<kiwi, 1=""></kiwi,>	<cherry, 1=""></cherry,>
Mapper 1 – outpu	t file	Mapper 3 – output file		Mapper 5 – output file	
<banana, 1=""></banana,>		<banana, 1=""></banana,>		<bandana, 1=""></bandana,>	
<cherry, 1=""></cherry,>		<banana, 1=""></banana,>		<cherry, 1=""></cherry,>	
<cherry, 1=""></cherry,>		<banana, 1=""></banana,>		<cherry, 1=""></cherry,>	
<cherry, 1=""></cherry,>		<cherry, 1=""></cherry,>		<kiwi, 1=""></kiwi,>	
<cherry, 1=""></cherry,>		<kiwi, 1=""></kiwi,>		<kiwi, 1=""></kiwi,>	
mango, 1>		<kiwi, 1=""></kiwi,>		<olive, 1=""></olive,>	
<olive, 1=""></olive,>		<mango, 1=""></mango,>		<olive, 1=""></olive,>	
<plu><plum, 1=""></plum,></plu>		<plu><plu><pre><pre><pre><pre><pre><pre><pre><pre< td=""><td></td><td><radish, 1=""></radish,></td><td></td></pre<></pre></pre></pre></pre></pre></pre></pre></plu></plu>		<radish, 1=""></radish,>	
Mapper 2–Input Split 2 –output		Mapper 4 – Input Split 4-		Mapper 6 – Input Split 6-	
wapper z-input s	piit 2 –output	output		output	
<cherry, 1=""></cherry,>	<carrot, 1=""></carrot,>	<apple, 1=""></apple,>	<radish, 1=""></radish,>	<banana, 1=""></banana,>	<olive, 1=""></olive,>
<banana, 1=""></banana,>	<banana, 1=""></banana,>	<mango, 1=""></mango,>	<kiwi, 1=""></kiwi,>	<radish, 1=""></radish,>	<cherry, 1=""></cherry,>
<radish, 1=""></radish,>	<mango, 1=""></mango,>	<carrot, 1=""></carrot,>	<banana, 1=""></banana,>	<plu><plu><pre><pre><pre><pre><pre><pre><pre><pre< td=""><td><banana, 1=""></banana,></td></pre<></pre></pre></pre></pre></pre></pre></pre></plu></plu>	<banana, 1=""></banana,>
<radish, 1=""></radish,>	<cherry, 1=""></cherry,>	<plu><plu><pre></pre></plu></plu>	<olive, 1=""></olive,>	<banana, 1=""></banana,>	<radish, 1=""></radish,>
Mapper 2 – output file		Mapper 4 – output file		Mapper 6 – output file	
<banana, 1=""></banana,>		<apple, 1=""></apple,>		<bandana, 1=""></bandana,>	
<banana, 1=""></banana,>		<banana, 1=""></banana,>		<banana, 1=""></banana,>	
<carrot, 1=""></carrot,>		<carrot, 1=""></carrot,>		<banana, 1=""></banana,>	
<cherry, 1=""></cherry,>		<kiwi, 1=""></kiwi,>		<cherry, 1=""></cherry,>	
<cherry, 1=""></cherry,>		<mango, 1=""></mango,>		<olive, 1=""></olive,>	
<mango, 1=""></mango,>		<olive, 1=""></olive,>		<plu><plum, 1=""></plum,></plu>	
<radish, 1=""></radish,>		<plu><plum, 1=""></plum,></plu>		<radish, 1=""></radish,>	
<radish, 1=""></radish,>		<radish, 1=""></radish,>		<radish, 1=""></radish,>	
Shuffle & Sort					
Reducer 1 input		Reducer 2 input		Reducer 3 input	
<apple, [1]=""></apple,>		<kiwi, [1,1,1,1,1]=""></kiwi,>			
<pre><banana, [1,1,1,1,1,1,1,1,1,1,1]=""></banana,></pre>		<mango, [1,1,1,1]=""></mango,>		<radish, [1,1,1,1,1,1]=""></radish,>	
<carrot, [1,1]=""></carrot,>		<oli><li><oli>(1,1,1,1,1)</oli></li></oli>			
<pre><cherry, [1,1,1,1,1,1,1,1,1,1]=""></cherry,></pre>		<plum, [1,1,1,1]=""></plum,>			



#### Reducer output is the same for all the cases:

Reducer 1 output	Reducer 2 output	Reducer 3 output
<apple, 1=""></apple,>	<kiwi, 5=""></kiwi,>	<radish, 6=""></radish,>
<banana, 11=""></banana,>	<mango, 4=""></mango,>	
<carrot, 2=""></carrot,>	<olive, 5=""></olive,>	
<cherry, 10=""></cherry,>	<plum, 4=""></plum,>	

## b) With combiner, no in mapper combining (assume that the combiner will work all the time)

Machine 1		Machine 2		Machine 3	
Mapper 1 – Input Split 1- output		Mapper 3 – Input Split 3- output		Mapper 5 – Input Split 5- output	
<pre><cherry, 1=""> <mango, 1=""> <olive, 1=""> <cherry, 1=""></cherry,></olive,></mango,></cherry,></pre>	<pre><plum, 1=""> <cherry, 1=""> <banana, 1=""> <cherry, 1=""></cherry,></banana,></cherry,></plum,></pre>	 <banana, 1=""> <kiwi, 1=""> <plum, 1=""><banana, 1=""></banana,></plum,></kiwi,></banana,>	<mango, 1=""> <cherry, 1=""> <kiwi, 1=""> <banana, 1=""></banana,></kiwi,></cherry,></mango,>	<pre><olive, 1="">   <banana, 1="">   <radish, 1="">   <kiwi, 1=""></kiwi,></radish,></banana,></olive,></pre>	<pre><cherry, 1="">   <kiwi, 1="">   <olive, 1="">   <cherry, 1=""></cherry,></olive,></kiwi,></cherry,></pre>
Combiner 1 output—saved locally as mapper1 output file		Combiner 3 output— saved locally as mapper3 output file		Combiner 5 output— saved locally as mapper 5 output file	
 <banana, 1=""> <cherry, 4=""> <mango, 1=""> <olive, 1=""> <plum, 1=""></plum,></olive,></mango,></cherry,></banana,>		 <banana, 3=""> <cherry, 1=""> <kiwi, 2=""> <mango, 1=""> <plum, 1=""></plum,></mango,></kiwi,></cherry,></banana,>		 <banana, 1=""> <cherry, 2=""> <kiwi, 2=""> <olive, 2=""> <radish, 1=""></radish,></olive,></kiwi,></cherry,></banana,>	
Mapper 2–Input Split 2 –		Mapper 4 – Input Split 4-		Mapper 6 – Input Split 6-	
output <cherry, 1=""> <banana, 1=""> <radish, 1=""> <radish, 1=""></radish,></radish,></banana,></cherry,>	<pre><carrot, 1="">   <banana, 1="">   <mango, 1="">   <cherry, 1=""></cherry,></mango,></banana,></carrot,></pre>	<pre>output <apple, 1=""> <mango, 1=""> <carrot, 1=""> <plum, 1=""></plum,></carrot,></mango,></apple,></pre>	<radish, 1=""> <kiwi, 1=""> <banana, 1=""> <olive, 1=""></olive,></banana,></kiwi,></radish,>	output <banana, 1=""> <radish, 1=""> <plum, 1=""> <banana, 1=""> <banana, 1=""></banana,></banana,></plum,></radish,></banana,>	<olive, 1=""> <cherry, 1=""> <banana, 1=""> <radish, 1=""></radish,></banana,></cherry,></olive,>
Combiner 2 output—saved locally as mapper2 output file		Combiner 4 output—saved locally as mapper4 output file		Combiner 6 output—saved locally as mapper6 output file	
 <banana, 2=""> <carrot, 1=""> <cherry, 2=""></cherry,></carrot,></banana,>		<apple, 1=""> <banana, 1=""> <carrot, 1=""></carrot,></banana,></apple,>		  <cherry, 1=""> <olive, 1=""></olive,></cherry,>	



<mango, 1=""></mango,>	<kiwi, 1=""></kiwi,>	<plum, 1=""></plum,>	
<radish, 2=""></radish,>	<mango, 1=""></mango,>	<radish, 2=""></radish,>	
	<olive,1></olive,1>		
	<pl><plum, 1=""></plum,></pl>		
	<radish, 1=""></radish,>		
Shuffle & Sort			
Reducer 1 input	Reducer 2 input	Reducer 3 input	
<apple, [1]=""></apple,>	<kiwi, [2,1,2]=""></kiwi,>		
 <banana, [1,2,3,1,1,3]=""></banana,>	<mango, [1,1,1,1]=""></mango,>	aradich [1 2 1 2]>	
	0 / [ / / / ]		
<carrot, [1,1]=""></carrot,>	<oli><li><oli>(1,1,2,1)</oli></li></oli>	<radish, [1,2,1,2]=""></radish,>	

#### c) With in-mapper combining

Machine 1	Machine 2	Machine 3		
Mapper 1 – Input Split 1-	Mapper 3 – Input Split 3-	Mapper 5 – Input Split 5-		
output file	output file	output file		
<banana, 1=""></banana,>	<banana, 3=""></banana,>	<banana, 1=""></banana,>		
<cherry, 4=""></cherry,>	<cherry, 1=""></cherry,>	<cherry, 2=""></cherry,>		
<mango, 1=""></mango,>	<kiwi, 2=""></kiwi,>	<kiwi, 2=""></kiwi,>		
<olive, 1=""></olive,>	<mango, 1=""></mango,>	<olive, 2=""></olive,>		
<plu><plum, 1=""></plum,></plu>	<plu><plum, 1=""></plum,></plu>	<radish, 1=""></radish,>		
Mapper 2-Input Split 2 -	Mapper 4 – Input Split 4-	Mapper 6 – Input Split 6-		
output file	output file	output file		
<banana, 2="">  <carrot, 1="">  <cherry, 2="">  <mango, 1="">  <radish, 2=""></radish,></mango,></cherry,></carrot,></banana,>	<apple, 1=""> <bannal <br=""></bannal>     <arrot, 1="">   <kiwi, 1=""> <mango, 1=""> <olive,1> <plum, 1=""> <radish, 1=""></radish,></plum,></olive,1></mango,></kiwi,></arrot,></apple,>	<banana, 3=""> <cherry, 1=""> <olive, 1=""> <plum, 1=""> <radish, 2=""></radish,></plum,></olive,></cherry,></banana,>		
Shuffle & Sort				
Reducer 1 input	Reducer 2 input	Reducer 3 input		
<apple, [1]=""> <banana, [1,2,3,1,1,3]=""> <carrot, [1,1]=""> <cherry, [4,2,1,2,1]=""></cherry,></carrot,></banana,></apple,>	<kiwi, [2,1,2]=""> <mango, [1,1,1,1]=""> <olive, [1,1,2,1]=""> <plum, [1,1,1,1]=""></plum,></olive,></mango,></kiwi,>	<radish, [1,2,1,2]=""></radish,>		

